Design Doc: Interact Online System

Summary:

This is a single page web application supports real-time online chatting, post letter, read letters and quick access to hot websites.

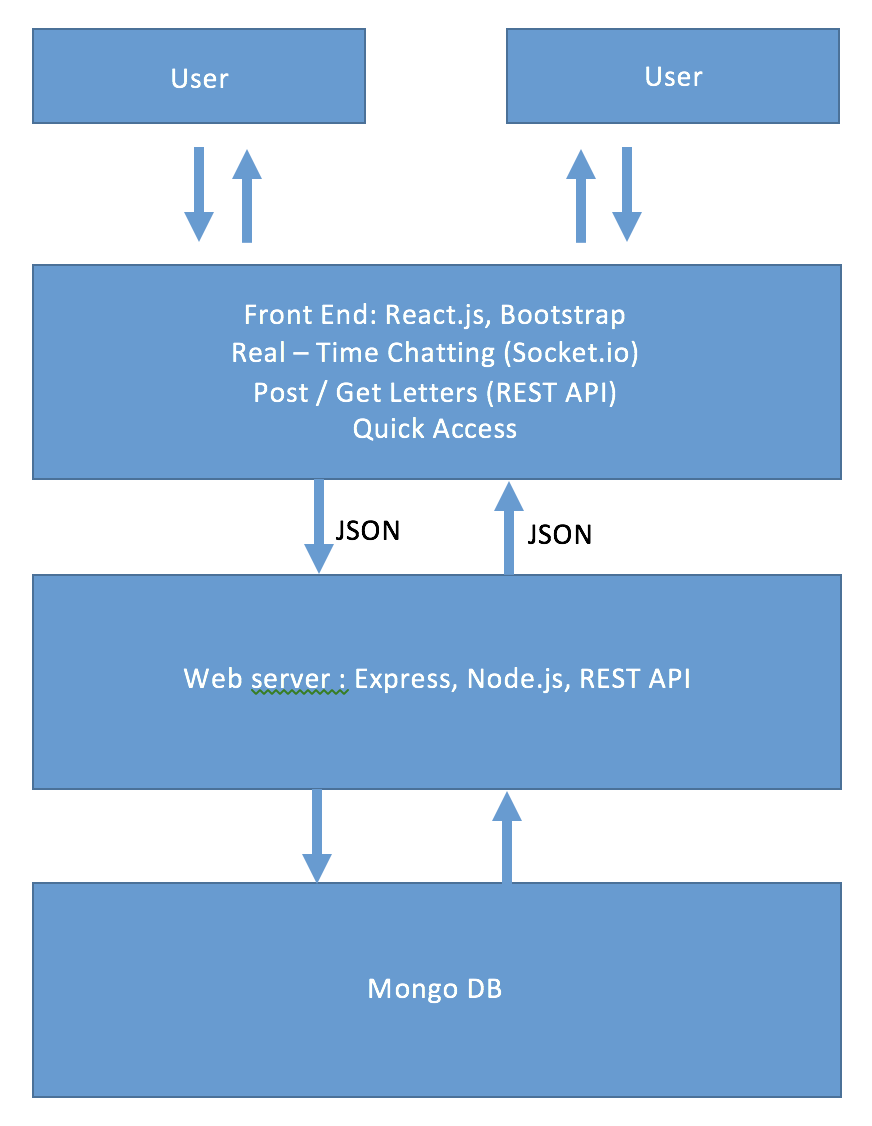
User Case:

1. Users can chat in real-time
2. Users can post new letters
3. Users can pick and read letters from letter pool
4. (Users can reply letter)
5. Users can quick access hot websites

Technology Stack

|  |  |
| --- | --- |
| Front End | React.js, socket.io, RESTful API |
| Back End | Express, Node.js, socket.io, RESTful API |
| Database | Redis, MongoDB |

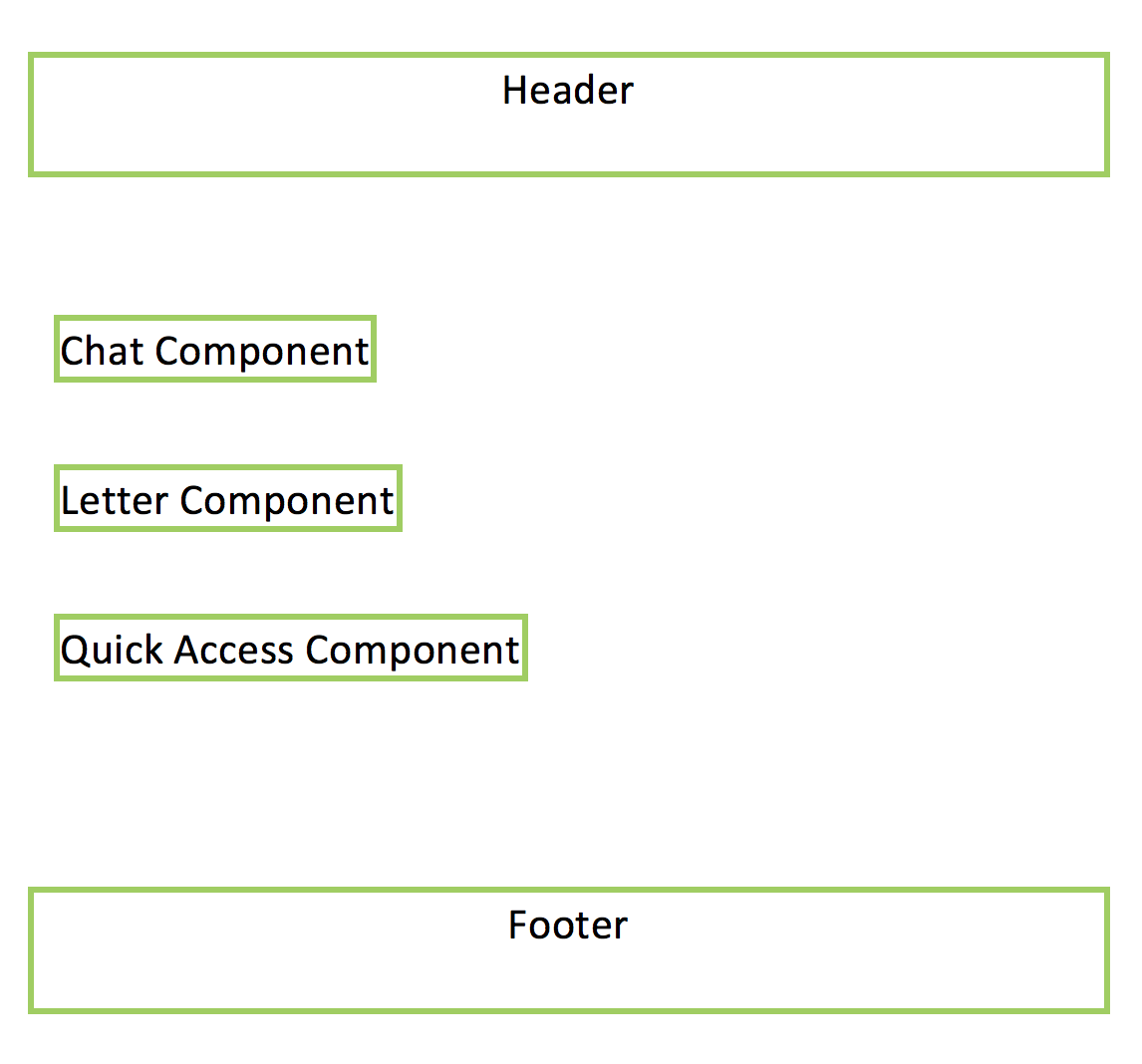
Data Flow Chart



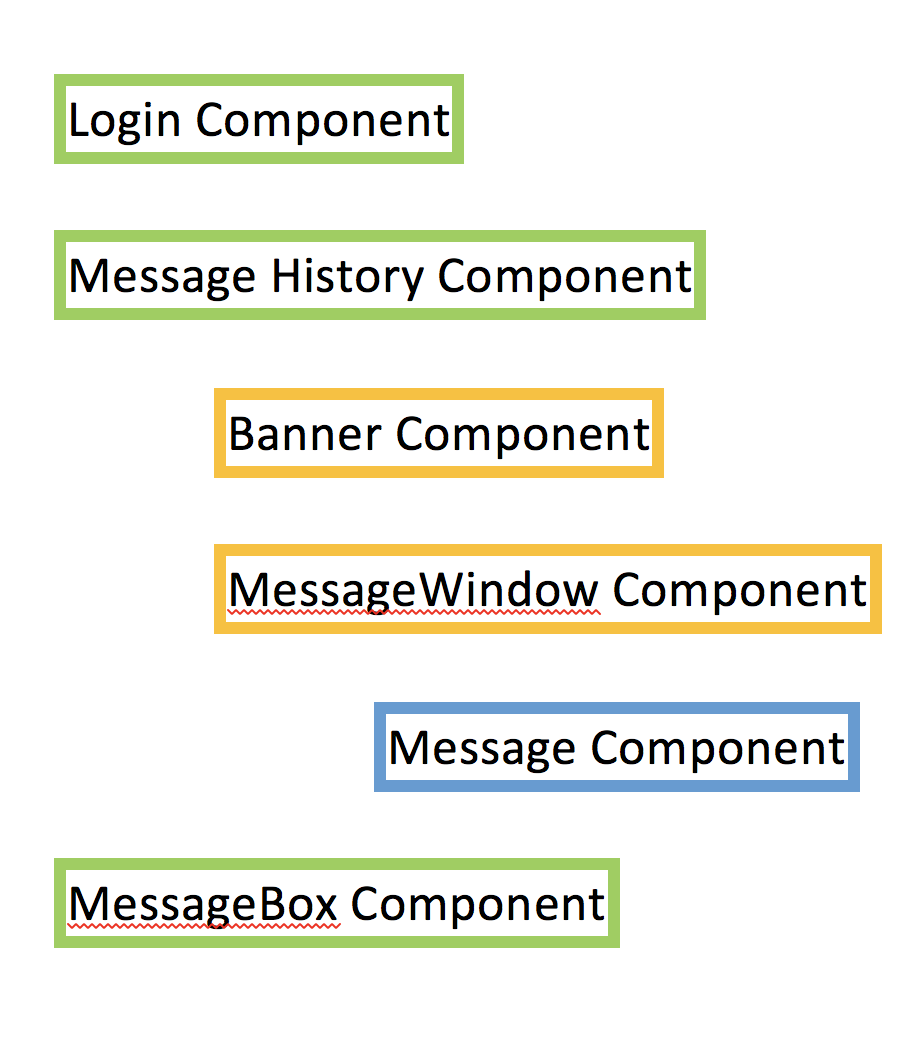
Implementation Details

1. UI presentation

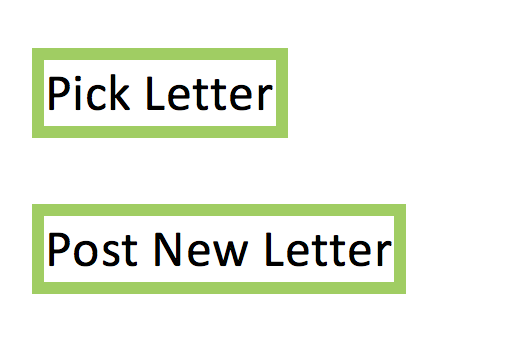
**Front Page**:



**Chat Component:**



**Letter Component**



**Quick Access Component**



2. Frontend

technology stack: React.js, socket.io, REST API, bootstrap

App component:

Entry point, route between chat component, letter component and quick access component

Letter component:

Use Restful APIs to talk with the backend, the APIs used: POST and GET.

POST: <http://localhost:3000/newletter> to post new letters to the server and database

GET: <http://localhost:3000/randomletter> to pick and read random letter from the serve

Chat component:

Use socket.io to support real-time communication between users

Consist of the following component:

1. Login component
2. Message history component
   1. Banner
   2. Message
3. Message box component

Quick Access component:

Support quick access to hot websites

3. Backend

technology stack: Express, Node.js, socket.io, RESTful API

Testing Plan

…

Scalability Plan

Nginx

Deployment Plan

Amazon AWS